

Bases de Datos

Orientación a los datos

- Los datos se almacenan en una **estructura lógica** basada en las características de los **propios datos**.
- El tratamiento de los datos se realiza mediante **aplicaciones y programas** que acceden a estas estructuras lógicas.
- El almacenamiento y la organización de los datos son **independientes** de su procesamiento.
- Varias aplicaciones y procesos **distintos** pueden acceder a las mismas estructuras de datos.
- Una base de datos **permitirá**:
 - Reunir toda la información relacionada en un **único sistema de almacenamiento**.
 - Cualquier aplicación podrá usarla de forma **independiente**.
 - Facilita el tratamiento de la información y la evolución para el **desarrollo de aplicaciones**.
- Base de datos: Estructura de datos interrelacionados a la que pueden acceder **simultáneamente** varios usuarios.

Ventajas

- **Independencia de los datos de los programas que los utilizan:** Permite modificar los datos sin necesidad de modificar el código de las aplicaciones.
- **Control de la redundancia:** No se necesitan repeticiones de datos.
- **Consistencia:** Menor riesgo de que el mismo dato tome valores distintos.
- **Integridad:** Menor riesgo de errores al manipular los datos.
- **Seguridad:** Permite controlar qué usuarios pueden acceder a los datos.
- **Acceso múltiple:** Diversos usuarios o aplicaciones podrán acceder simultáneamente a la base de datos.
- **Confidencialidad y seguridad:** Se puede establecer un control de acceso a los datos para que algunos usuarios o aplicaciones puedan acceder a algunos datos y otros no, evitando que usuarios no autorizados utilicen la base de datos.
- **Documentación:** Incluye metadatos que describen la información de la base de datos (qué datos hay, qué representan, cuáles son sus posibles valores, qué dependencias tienen con otros datos, quién puede acceder, etc.).
- **Eficiencia:** Una mejor organización de los datos reduce el tiempo necesario para procesarlos.
- **Menos almacenamiento:** Una mejor estructuración de los datos, sin duplicados, reduce el espacio de almacenamiento necesario.

Inconvenientes

- **Coste:** El software necesario para el almacenamiento de los datos y su gestión puede ser costoso.
- Requiere **personal especializado**.
- **Implementación larga y difícil.**
- **Ausencia de estándares reales:** Los programas de gestión comerciales introducen pequeñas variaciones que se alejan del estándar.

Visión abstracta de los datos

- En los sistemas basados en BD:
 - El almacenamiento se orienta hacia las **características de los datos**.
 - **Programas y aplicaciones** acceden a los datos para procesarlos.
- Visión abstracta de los datos: Los **programas** que utilizan la BD **no tienen que conocer** detalles sobre el almacenamiento y mantenimiento de los datos:
 - **Facilidad de uso:** Solo deben centrarse en sus procesos.
 - **Seguridad:** Solo tendrán acceso a lo que necesiten.
- Arquitectura de tres niveles:
 - Comité ANSI-SPARC (1975)
 - Objetivo: **separar** los programas de aplicación de la base de datos física.
 - 3 niveles de abstracción:
 - **Físico o interno:** Describe cómo se almacenan físicamente las estructuras de datos.
 - **Conceptual o lógico:** Describe la organización lógica de los datos y sus relaciones.
 - **Externo de usuario o de vista:** Describe la BD tal como la perciben los usuarios.
- Nivel físico o interno:
 - Describe **cómo se almacenan físicamente** las **estructuras** de datos. Especifica: los archivos que contienen la información, su organización, los métodos de acceso a los registros, los tipos de registros, la longitud, los campos que los componen, las rutas de acceso, etc.
 - La descripción del nivel físico se hace a través de un **esquema interno**: un conjunto de definiciones y reglas que permite definir las tablas y cómo están relacionadas entre sí.
 - Los usuarios que trabajan en este nivel son los **diseñadores o administradores** de bases de datos. Es raro trabajar en este nivel. Solo se hace cuando se necesitan optimizaciones de estructuración de datos de bajo nivel.

- Nivel conceptual o lógico:
 - Describe la **organización lógica** de los y las relaciones entre ellos. Trata de la estructura organizativa de los datos sin abordar las estructuras de almacenamiento físico.
 - La descripción del nivel conceptual se hace a través de un **esquema conceptual**: define las entidades, los atributos y sus propiedades, las relaciones, las operaciones del usuario y las reglas y restricciones de validación.
 - Los usuarios que trabajan en este nivel son los **programadores y administradores** de bases de datos, encargados de crear las estructuras lógicas necesarias para guardar la información.
- Nivel externo o de usuario o de vista:
 - Describe la base de datos tal como la **perciben los usuarios**.
 - La mayoría de los usuarios no necesitan conocer toda la estructura lógica de la base de datos.
 - Los usuarios no tienen que conocer detalles de la complejidad interna.
 - La descripción del nivel externo se hace a través del **esquema externo**: conjunto de objetos de la base de datos a los que puede acceder un usuario, como tablas, vistas, formularios, informes, etc.
 - El nivel externo es la percepción de la base de datos que tiene un **usuario**; hay tantos niveles externos diferentes como grupos de usuarios.

Independencia de los datos

- Arquitectura de tres niveles ANSI-SPARC y el concepto de independencia de datos.
- 3 niveles de abstracción conducen a 2 tipos de independencia.
- 2 tipos de independencia:
 - **Independencia física**: Posibilidad de modificar el esquema interno sin tener que modificar ni el esquema conceptual ni el esquema externo.
 - **Independencia lógica**: Posibilidad de modificar el esquema conceptual de la base de datos sin tener que modificar los esquemas externos ni los programas.
- **Independencia física**: Posibilidad de modificar el esquema interno sin tener que modificar ni el esquema conceptual ni el esquema externo. Por ejemplo, no afectaría a los modelos lógicos ni a las aplicaciones:
 - Reorganizar archivos físicos o añadir nuevos archivos de datos para mejorar el rendimiento.
 - Cambiar los datos de un soporte físico a otro.

- Cambiar la forma de acceso a los registros (por ejemplo, usar un Btree en lugar de un índice).
- Cambiar el formato de la codificación.
- ...
- **Independencia Lógica:** Posibilidad de modificar el esquema conceptual de la base de datos sin tener que modificar los esquemas externos ni los programas. Por ejemplo, no afectaría a las vistas ni a las aplicaciones:
 - Eliminar un atributo de una entidad (excepto a las que usen ese atributo).
 - Eliminar una entidad (excepto a las que la usen).

Modelos de bases de datos

- **Modelo:** Representación de una parte del mundo real.
 - Debe incluir aquellos **aspectos importantes** de los elementos a representar y excluir detalles de bajo nivel o irrelevantes.
 - Su observación y manipulación debe permitir obtener los mismos resultados que los que se obtendrían en la realidad.
- **Proceso de modelado:** Definir una representación a partir de una parte del mundo real.
- **Modelo de datos:** Conjunto de **herramientas conceptuales** que permiten describir datos, sus relaciones y las reglas de integridad que debe cumplir.
- El modelo de datos se establece durante el diseño de la base de datos y generalmente no cambia mucho. Los datos almacenados en la base de datos, en cambio, suelen cambiar con más frecuencia.
- **Evolución y tipos de modelos de datos:**
 - Jerárquicos (principios de los 60).
 - En red (finales de los 60).
 - Relacional (desde 1970, es el más usado actualmente).
 - Multidimensionales.
 - Relacional extendido (evolución del relacional).
 - De objetos (para sistemas orientados a objetos y multimedia).

Modelo Jerárquico

- **Primeras bases de datos** (principios de los 60).
- Los datos se organizan en registros almacenados **jerárquicamente** en forma de **árbol invertido**.
- Los registros se llaman segmentos o nodos y contienen **atributos** o **campos**.
- Los nodos están organizados en niveles:
 - Cada nodo puede tener **un solo nodo padre** y **varios nodos hijos**.
 - El nodo en el nivel más alto se llama **raíz** y no tiene padre.
- Cada nodo contiene los campos comunes a los nodos hijos, vinculados a él.

- Las relaciones entre registros están representadas por **arcos o lazos**.
- Fueron los modelos de bases de datos más usados en las décadas de **1970 y 1980**.
- Actualmente se utilizan **IMS de IBM** y **Registro de Datos de Windows** de Microsoft.

Modelo en red

- **Evolución de la jerárquica** (finales de los 60).
- Inventada por **Charles Bachman** en 1969 y adoptada por **CODASYL**.
- Mantiene las relaciones jerárquicas, pero **cada registro puede tener más de un padre**: los registros se relacionan en forma de red.
- Permite representar **cualquier relación** entre los datos.
- El manejo de los datos es **complejo**.
- Utilizada, por ejemplo, por **IDMS** (Sistema de Gestión de Bases de Datos Integradas) de **Computer Associates**.

Modelo Relacional

- Modelo relacional de **Codd** (1970):
 - Plantea que los datos deberían relacionarse a través de **interrelaciones naturales y lógicas**.
 - Los datos se representan como tablas llamadas **relaciones**.
 - Cada fila de la tabla representa un **registro**.
 - Cada columna de la tabla representa un **campo**.
- Las **tablas son independientes** pero están relacionadas por una **vinculación común**.
- Permite **redundancia e inconsistencia mínimas**.
- Los datos pueden ser manejados aplicando **álgebra y cálculo relacional**.
- **Ventajas:**
 - Los usuarios y diseñadores perciben el sistema como un conjunto de tablas, los **detalles físicos complejos permanecen ocultos**.
 - **Independencia entre los datos y los dispositivos** de almacenamiento: los detalles de almacenamiento permanecen ocultos.
 - **Independencia estructural:** Los usuarios se centran en los datos a los que deben acceder, y no en la ruta que deben seguir para acceder a ellos. Cambios en la estructura de la base de datos no afectan a la capacidad de acceso a los datos.
 - **Sencillez conceptual:** debido a que el sistema cuida el almacenamiento físico de los datos, los diseñadores se centran en la representación lógica de la base de datos.
 - **Facilidad de diseñar, gestionar y utilizar** la base de datos, debido a la independencia estructural y de los datos.

- Permite realizar consultas y manipulaciones con un lenguaje sencillo (lenguaje **SQL**).
- **Inconvenientes:**
 - Requiere una **inversión considerable** en hardware y software para evitar que sea lento (cada vez menos gracias a la evolución del hardware y los sistemas operativos).
 - El **diseño deficiente es bastante común** debido a la facilidad de uso de esta herramienta para personas sin experiencia. A medida que la base de datos crece, si el diseño es inadecuado, el sistema se vuelve más lento y se producen anomalías.
 - Debido a la facilidad de uso, los usuarios finales tienden a crear **subconjuntos** de bases de datos que pueden generar **datos inconsistentes**.
- El modelo relacional reemplazó al modelo de red porque:
 - Permite representar la realidad de una manera más **sencilla**.
 - Ofrece una mayor **independencia** de datos.
 - Admite consultas y manipulaciones con un **lenguaje relativamente sencillo**.
- Es el **modelo más utilizado** actualmente.
- Evolución: modelo **relacional ampliado** y modelos **orientados a objetos** (más visuales).
- Se estudiará con más detalle en las siguientes unidades.

Modelo Multirrelacional

- Variante de la base de datos relacional en la que los datos se organizan en **más de dos dimensiones** (no solo filas y columnas).
- Tiene cierta similitud con las **hojas de cálculo**.
- Apenas se utiliza.

Modelo orientado a objetos

- Diseñada según el paradigma de los **lenguajes orientados a objetos** (años 90).
- Los datos se representan en **forma de objetos**.
- Cada objeto almacena:
 - Los propios **datos**.
 - Las **relaciones** con otros objetos.
 - Las **operaciones** que se pueden realizar con esos datos.
- La estructura básica es el **objeto**: representa una **entidad individual** (el cliente Frinsa, el alumno Miguel, ...).
- Los objetos se agrupan en **clases** que los definen:

- Todos los objetos de la **misma clase** tendrán las **mismas características: atributos**.
- Todos los objetos de la **misma clase** tendrán los **mismos comportamientos: métodos**.
- Ejemplo: Clase alumno:
 - Atributos: nombre, dirección, materias matriculadas, ...
 - Métodos: matricularse, convalidar, aprobar, ...
- Las clases se organizan en una **jerarquía de clases** que se asemeja a un árbol invertido donde cada clase tiene solo un padre.
- Ejemplo: la clase cliente y la clase proveedor pertenecen a la clase padre empresa.
- La **herencia** es la capacidad de un objeto para heredar los **atributos y métodos** de sus antecesores.
- Ejemplo: la clase cliente y la clase proveedor heredan los atributos y métodos de la clase empresa.
- Soporta tipos de **datos gráficos, imágenes, voz y texto** de forma natural.
- Suele utilizarse en **aplicaciones web** para multimedia.